New formulations for the Kissing Number Problem

Leo Liberti, Pietro Belotti, Sergei Kucherenko, Nelson Maculan

New formulations for the Kissing Number Problem

Sergei Kucherenko

CPSE, Imperial College, SW7 2AZ, London, UK
s.kucherenko@imperial.ac.uk

Pietro Belotti

DEI, Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy
belotti@elet.polimi.it

Leo Liberti

LIX, École Polytechnique, Palaiseau F-91128, France
liberti@lix.polytechnique.fr

Nelson Maculan

COPPE, Universidade Federal do Rio de Janeiro, P.O. Box 68511, 21941-972 Rio de Janeiro, Brazil
maculan@cos.ufrj.br

Abstract

Determining the maximum number of D-dimensional spheres of radius r that can be adjacent to a central sphere of radius r is known as the Kissing Number Problem (KNP). The problem has been solved for 2, 3 and very recently for 4 dimensions. We present two nonlinear (nonconvex) mathematical programming models for the solution of the KNP. We solve the problem by using two stochastic global optimization methods: a Multi Level Single Linkage algorithm and a Variable Neighbourhood Search. We obtain numerical results for 2, 3 and 4 dimensions.

Keywords: sphere packing, NLP, global optimization, stochastic algorithm, multi-level single linkage, variable neighbourhood search.

1 Introduction

When rigid balls touch each other, in billiard-room terms, they "kiss". In mathematical terms, the *kissing number* in D dimensions is the number of D-

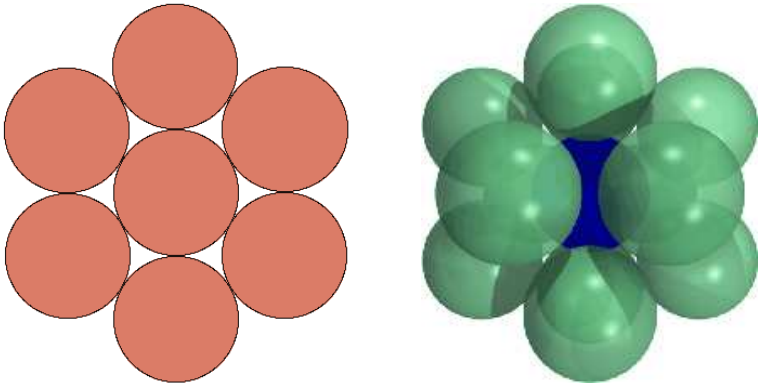

Fig. 1. The problem in $\mathbb{R}^2$ (a) and $\mathbb{R}^3$ (b). Figure (b) was taken from [18].

spheres of unit radius that can be arranged around a central D-sphere of unit radius so that each of the surrounding spheres touches the central one without overlapping. Determining the maximum kissing number in various dimensions has become a well-known problem in Combinatorial Geometry. Notationally, we indicate the Kissing Number Problem in D dimensions by KNP(D).

In $\mathbb{R}^2$ the result is trivial: the maximum kissing number is 6 (Fig. 1, a). The situation is far from trivial in $\mathbb{R}^3$. The problem earned its fame because, according to Newton, the maximum kissing number in 3D was 12, whereas according to his contemporary fellow mathematician David Gregory, the maximum kissing number in 3D was 13 (this fact was stated without proof). This question was settled, at long last, more than 250 years after having been stated, when J. Leech finally proved that the solution in 3D is 12 [7]. The question for the 4-dimensional case was very recently settled in a yet unpublished paper by O. Musin of Moscow State University [11] which shows that the solution of KNP(4) is 24 spheres. In this paper, we propose a mathematical programming approach based on two nonconvex continuous models, which we solve with two global optimization algorithms for the 2, 3 and 4 dimensional cases. One of these algorithms is a quasi Monte Carlo variant of the Multi Level Single Linkage (MLSL) algorithm called SobolOpt [6], the other is a novel implementation of Variable Neighbourhood Search for constrained NLPs. Previously, the KNP had been used as a validation method for a local optimization code [10,5].

2 The models

We propose two models both of which rely on a variable being maximized (or minimized) with an associated threshold, called the *feasibility indicator*. If the globally optimal value of the feasibility indicator is higher than the threshold, an N-sphere packing in D dimensions is feasible, otherwise it is infeasible. Using a bisection principle, we can quickly pinpoint the optimal solution N^* to KNP(D) as the minimum N such that the $N+1$ packing is infeasible.

We propose two formulations for the KNP. The first one, which we call the "distance formulation", is a special case of a more general formulation found in the technical report [9]. Given parameters D (dimension of Euclidean space) and N (number of spheres in the packing), the variables $x^i = (x_1^i, \ldots, x_D^i)$, $1 \leq i \leq N$ determine the position of the center of the i-th sphere around the central one. We maximize the feasibility indicator $\alpha \geq 0$ (with threshold 1), which represents the minimum pairwise sphere separation distance in the N-sphere configuration being tested, subject to the necessary geometric constraints. Since the constraints are nonconvex, there may be multiple local minima. If the solution is $\alpha \geq 1$, then there is enough space for N spheres, otherwise there are overlapping spheres and $N > N^*$.

$$\begin{aligned}
\max \quad & \alpha && (1)\\
\forall i \leq N \quad & ||x^i||^2 = 4 && (2)\\
\forall i < j \leq N \quad & ||x^i - x^j||^2 \geq 4\alpha && (3)\\
& \alpha \geq 0 && (4)\\
\forall i \leq N \quad & x^i \in \mathbb{R}^D, && (5)
\end{aligned}$$

where the norm $||\cdot||$ is taken to be the Euclidean norm. Constraints (2) ensure that the centers of the N spheres all have distance 2 from the center of the central sphere (i.e., the N spheres "kiss" the central sphere). Constraints (3) makes the N spheres non-overlapping. In practice we perform a simplification based on the following:

$$\begin{aligned}
||x^i - x^j||^2 &= \sum_{k=1}^{D} (x_k^i - x_k^j)^2 = \sum_{k=1}^{D} ((x_k^i)^2 + (x_k^j)^2 - 2x_k^i x_k^j) = \\
&= ||x^i||^2 + ||x^j||^2 - 2\sum_{k=1}^{D} x_k^i x_k^j.
\end{aligned}$$

By Eq. (2) we have $||x^i - x^j||^2 = 8 - 2\sum_{k=1}^{D} x_k^i x_k^j$, thus constraint (3) can be reformulated to

$$\forall i < j \leq N \quad 2\alpha + \sum_{k=1}^{D} x_k^i x_k^j \leq 4. \qquad (6)$$

This reduces the number of nonlinear terms in the problem and makes it somewhat faster to solve in a local stage of a global optimization algorithm.

We propose a second formulation, which we call the "angle formulation". As all centres have distance 2 from the centre of the central sphere, their position is uniquely identified by their spherical coordinates ϑ_k^i, for $k \in \{1, 2, \ldots, D -$

1} and $i \in \{1, 2, \ldots, N\}$. Adopting spherical coordinates ϑ_k^i, the cartesian coordinates x_i^k of the sphere centers are defined as

$$x_k^i = \cos \vartheta_k^i \prod_{h=1}^{d-k} \sin \vartheta_h^i.$$

We obtain a formulation with polynomial constraints by considering the sines and cosines of each ϑ_k^i defined as variables $\sigma_k^i = \sin \vartheta_k^i$ and $\gamma_k^i = \cos \vartheta_k^i$, respectively. The angle formulation can be defined as follows (we assume $\gamma_D^i = 1$ for each $i \leq N$):

$$\begin{aligned}
\min \quad & \alpha && (7)\\
\forall i \leq N, k < D \quad & (\sigma_k^i)^2 + (\gamma_k^i)^2 = 1 && (8)\\
\forall i < j \leq N \quad & \sum_{k=1}^{D} \left(\gamma_k^i \gamma_k^j \prod_{h=1}^{D-k} \sigma_h^i \sigma_h^j \right) \leq 0.5 + \alpha && (9)\\
\forall i \leq N \quad & \sigma^i \in [-1, 1]^{D-1}, \gamma^i \in [-1, 1]^D. && (10)
\end{aligned}$$

Constraints (8) impose that $\sin^2 \vartheta_k^i + \cos^2 \vartheta_k^i = 1$, while constraints (9) require that the angle between any two vectors must not exceed $\frac{\pi}{3}$ (which implies that the spheres will not overlap), where the feasibility indicator α is employed in the minimization direction. If for a given (N, D) the globally optimal α is zero, then we have found a sphere packing.

3 The methods

A stochastic approach for global optimization, in its simplest form, consists only of random search and it is called Pure Random Search (PRS). In PRS an objective function $f(x)$ is evaluated at P randomly chosen points and the smallest value of $f(x)$ is taken as the global minimum. Advanced stochastic techniques use stochastic methods to search for local minima and then utilize deterministic methods to solve a local minimisation problem. Two phases are considered: global and local. In the global phase, the function is evaluated in a number of randomly sampled points from a uniform distribution over a unit Hypercube H_n. In the local phase the sample points are used as starting points for a local minimization search. The efficiency of the multistage methods depends both on the performance of the global stochastic and the local minimization phases.

In the most basic form of the multistage approach a local search is applied to every sample point. Inevitably, some local minima would be found many

times. Since the local search is the most CPU-time consuming stage, ideally it should start just once in every region of attraction. This is the idea behind various versions of the so-called clustering methods. Extensive reviews on this subject can be found in [17,15]. One of the most efficient clustering methods is a MLSL algorithm developed by Rinnooy-Kan and Timmer in [13,14].

The efficiency of MLSL depends on the quality of sampled points. It has been recognized through theory and practice that uniformly distributed deterministic sequences provide more accurate results than purely random sequences. Low-discrepancy sequences (LDS) are designed specifically to place sample points as uniformly as possible. Unlike random numbers, successive low discrepancy points "know" about the position of their predecessors and fill the gaps left previously. Methods based on LDSs are known as quasi Monte Carlo (QMC) methods. In the majority of applications, QMC methods have superior performance compared to that of MC methods. Improvement in time-to-accuracy using QMC methods can be as large as several orders of magnitude. It was shown in [6] that application of LDS can significantly increase the efficiency of MLSL methods. Central to the QMC approach is the choice of LDS. Different principles were used for constructing LDSs by Holton, Faure, Sobol', Niederreiter and others. Many practical studies have proven that Sobol' LDS in many aspects are superior to other LDSs [12,16].

Variable Neighbourhood Search (VNS) is a relatively recent metaheuristic method which relies on iteratively exploring neighbourhoods of growing size to identify better local optima [3,4]. More precisely, VNS escapes from the current local minimum x^* by initiating other local searches from starting points sampled from a neighbourhood of x^* which increases its size iteratively until a local minimum better than the current one is found. These steps are repeated until a given termination condition is met. The search space is defined as the hyper-rectangle given by the set of variable ranges $x^L \leq x \leq x^U$. For each $k \leq k_{\max}$ we define hyper-rectangular neighbourhoods $N_k(x^*)$ with side lengths proportional to those of $x^L \leq x \leq x^U$, centered at x^*, whose sides have been scaled by $\frac{k}{k_{\max}}$. More precisely, we let $N_k(x^*)$ be the hyper-rectangle $y^L \leq x \leq y^U$ where, for all $i \leq n$:

$$y_i^L = x_i^* - \frac{k}{k_{\max}}(x_i^* - x_i^L), \qquad y_i^U = x_i^* + \frac{k}{k_{\max}}(x_i^U - x_i^*).$$

4 Computational results

We solved the KNP by using solvers implementing the methods above within the framework of the general-purpose global optimization software framework

ooOPS [8]. Both solvers call a local constrained NLP solver code (SNOPT [2]) to perform the local descent.

Our computational results were all obtained with a 800MHz Intel Pentium III CPU with 384 MB RAM running Linux. In Table 1 we report results from the first model; n, m are the number of variables and constraints in the problem, rows SobolOpt and VNS report the globally optimal feasibility indicator values and user CPU times (in seconds) taken by the SobolOpt and VNS solvers to globally solve the problem. Each column corresponds to a different (N, D) pair. In 2D, 6 spheres are a feasible packing, and 7 are not; in 3D, 12 spheres are feasible and 13 are not; in 4D, 24 and 25 are known lower and upper bounds to the KNP [1]. Table 2 contains the corresponding results for the second model. For each instance and solver, the correct feasibility indicator values were identified.

Threshold: 1	$D = 2$		$D = 3$		$D = 4$	
Max α	$N = 6$	$N = 7$	$N = 12$	$N = 13$	$N = 24$	$N = 25$
n	13	15	37	40	97	101
m	21	28	78	91	300	325
SobolOpt: α	1.00000	0.75302	1.10558	0.91473	1.00000	0.92537
(CPU)	5.05	6.24	31.35	40.72	334.36	369.55
VNS: α	1.00000	0.75302	1.10558	0.9147	1.00000	0.92537
(CPU)	0.31	0.44	1.88	3.39	30.07	24.35

Table 1
Computational results for the distance model (1)-(5). CPU is in seconds of user time.

Threshold: 0	$D = 2$		$D = 3$		$D = 4$	
Min α	$N = 6$	$N = 7$	$N = 12$	$N = 13$	$N = 24$	$N = 25$
n	13	15	37	40	97	101
m	21	28	90	104	348	375
SobolOpt: α	0.00000	0.12349	0.00000	0.04263	0.00000	0.03731
(CPU)	2.75	5.32	120.58	215.25	4861.53	5565.55
VNS: α	0.00000	0.12349	0.00000	0.04263	0.00000	0.03731
(CPU)	0.62	0.69	9.76	20.56	850.22	816.23

Table 2
Computational results for the angle model (7)-(10). CPU is in seconds of user time.

Our results show that the distance formulation is more promising, in terms of user CPU time, than the angle formulation, for both SobolOpt and VNS global solvers.

5 Conclusion

In this paper two nonconvex continuous mathematical programming formulations to solve the decision problem "is N the Kissing Number in Euclidean D-space?" were presented. The Kissing Number was correctly determined in 2, 3 and 4 dimensions, using two different stochastic global optimization algorithms. Research is ongoing for obtaining computational results for the smallest open case, namely 5 dimensions.

References

[1] J. Conway, N. Sloane (Eds.), Sphere Packings, Lattices and Groups, Springer-Verlag, Berlin, 1993.

[2] P. Gill, User's Guide for SNOPT 5.3, Systems Optimization Laboratory, Department of EESOR, Stanford University, California, 1999.

[3] P. Hansen, N. Mladenović, Variable neighbourhood search, in: F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer, Dordrecht, 2003.

[4] P. Hansen, N. Mladenović, Variable neighbourhood search: Principles and applications, European Journal of Operations Research 130 (2001) 449–467.

[5] N. Krejič, J. Martínez, M. Mello, E. Pilotta, Validation of an augmented lagrangian algorithm with a gauss-newton hessian approximation using a set of hard-spheres problems, Computational Optimization and Applications 16 (3) (2000) 247–263.

[6] S. Kucherenko, Y. Sytsko, Application of deterministic low-discrepancy sequences to nonlinear global optimization problems, Computational Optimization and Applications 30 (3) (2004) 297–318.

[7] J. Leech, The problem of the thirteen spheres, Mathematical Gazette 40 (1956) 22–23.

[8] L. Liberti, P. Tsiakis, B. Keeping, C. Pantelides, *oo*$\mathcal{OPS}$, Centre for Process Systems Engineering, Chemical Engineering Department, Imperial College, London, UK (2001).

[9] N. Maculan, P. Michelon, J. MacGregor Smith, Bounds on the kissing numbers in $\mathbb{R}^n$: Mathematical programming formulations, Technical Report, University of Massachusetts, Amherst, USA, 1996.

[10] J. Martínez, Augmented lagrangians and sphere packing problems, Technical Report, University of Campinas, Brazil, 1997.

[11] O. Musin, The kissing number in four dimensions, Technical Report, Moscow State University, 2004.

[12] S. Paskov, J. Traub, Faster evaluation of financial derivatives, Journal of Portfolio Management 22 (1) (1995) 113–120.

[13] A. Rinnooy-Kan, G. Timmer, Stochastic global optimization methods; part I: Clustering methods, Mathematical Programming 39 (1987) 27–56.

[14] A. Rinnooy-Kan, G. Timmer, Stochastic global optimization methods; part II: Multilevel methods, Mathematical Programming 39 (1987) 57–78.

[15] F. Schoen, Two-phase methods for global optimization, in P.M. Pardalos, E. Romeijn, (Eds.), Handbook of Global Optimization vol. 2, Kluwer, Dordrecht, 2002, 151–177.

[16] I. Sobol', On quasi-monte carlo integrations, Mathematics and Computers in Simulation 47 (1998) 103–112.

[17] A. Törn, A. Žilinskas, Global Optimization, Springer-Verlag, Berlin, 1989.

[18] E.W. Weisstein, Kissing Number, MathWorld–A Wolfram Web Resource, `http://mathworld.wolfram.com/KissingNumber.html`.